Eötvös Loránd Tudományegyetem

Informatikai Kar

Média- és Oktatásinformatikai

Tanszék

Közlekedésieszköz-kölcsönző webalkalmazás turisták számára

Témavezető:
Bende Imre
egyetemi tanársegéd

Szerző:
Siklósi Gergely
programtervező informatikus BSc

Budapest, 2020

Az eredeti szakdolgozati / diplomamunka témabjelentő helye.

Tartalomjegyzék

1. fejezet

Bevezetés

Az utazás egyre nagyobb tényezővé válik a mindennapi életben. Egyre több ember választja a turizmus nyújtotta kikapcsolódást, felfedezést. Fontosnak tartom, hogy ha valaki meglátogatja környékünket, akkor színvonalas élményben legyen része. Ezért gondoltam, hogy egy közlekedési eszköz bérlő alkalmazás fejlesztése hasznos lehet. Az elképzelésem az, hogy a turisták az egyes kijelölt helyeken, például információs pontokon gyorsan, egyszerűen, zökkenőmentesen tudjanak kibérelni eszközöket, melyek segítik őket a gyors közlekedésben a környéken. Egy egyszerű bejelentkezés után már azonnal bérelhetnek is járműveket. A rendszer nyilvántartja a bérlés kezdetét, és amikor a vásárló visszaszolgáltatja a járművet, előállít egy számlát az igénybe vett szolgáltatásról, amit a bérlő a helyszínen kifizet. Nem csak turistáknak, de a cégeknek is egyszerű a felület használata. Cég esetében több joggal rendelkezik a felhasználó. Egy cég hirdethet eszközöket, vissza is veheti őket, ha valami hiba történne a bérlő személy részéről. A cégek hozzáférnek az összes még folyamatban lévő és már befejezett bérlésekhez, és bármikor letölthetik a számlát, akár visszamenőleg is. A program fejlesztése során a backend-et Java (Spring keretrendszer) segítségével valósítottam meg, adatbázist használva, a frontend-et TypeScript (Angular 2+) nyelven implementáltam. A felhasználói dokumentációban bemutatom, hogy a regisztrált felhasználók milyen funkciókat érhetnek el az oldalon, jogosultságaiknak megfelelően. A fejlesztői dokumentációban bemutatom a fejlesztés során használt eszközöket, az adatbázis, és a backend felépítését, tesztelését.

2. fejezet

Felhasználói dokumentáció

2.1. Az alkalmazás célja

Az alkalmazás egy város (például Budapest) turisztikai központjának készült. Az alkalmazás lehetővé teszi felhasználók regisztrálását. Három jogosultság támogatott: Guest - ezek a turisták; Company - ezek a cégek; Admin - ezek a felhatalmazott személyek, akik mindenki felett állnak. A Guest jogosultsággal rendelkező felhasználók bejelentkezés után kilistázhatják az aktuálisan bérelhető járműveket, bérelhetnek járműveket, az egyes közlekedési eszközöket hozzáadhatják kedvenceikhez, üzenetet küldhetnek a gépjármű tulajdonosoknak, bérlés után visszaszolgáltathatják az adott járművet, értékelhetik, hogy mennyire voltak megelégedve az igénybe vett szolgáltatással, ami a cég profilján megjelenik, és végül kifizethetik a bérlés költségét. A Company jogosultsággal rendelkező felhasználók bejelentkezés után kilistázhatják az aktuálisan bérelhető járműveket, bérelhetnek járműveket, az egyes közlekedési eszközöket hozzáadhatják kedvenceikhez, üzenetet küldhetnek a gépjármű tulajdonosoknak, bérlés után visszaszolgáltathatják az adott járművet, értékelhetik, hogy mennyire voltak megelégedve az igénybe vett szolgáltatással, ami a cég profilján megjelenik, kifizethetik a bérlés költségét. Ezen felül a saját hirdetéseiket módosíthatják, valamint törölhetik. Az Admin jogosultsággal rendelkező egyéneknek lehetőségük van bejelentkezés után kilistázni a bérelhető járműveket, üzenetet küldeni a gépjármű tulajdonosoknak, törölni hirdetéseket, kilistázni az eddigi bérléseket és számlát letölteni ezekről, kilistázni a felhasználókat és kitörölni azokat. Az alkalmazás célja, hogy a beregisztrált felhasználóknak gyors és egyszerű hozzáférést nyújtson

az alkalmazásban szereplő termékek bérléséhez.

2.1.1. Célközönség

Az alkalmazás turisták számára készült. Ez azt jelenti, hogy bárki, aki turizmus céljából látogatta meg a várost igénybe veheti. Azonban a célközönség nem korlátozódik le csak a turisták körére. Mivel az eszközök bérlésének a célja, hogy a városlátogatást könnyítse, ezért akár a város lakossága is ugyanúgy igénybe veheti az alkalmazás nyújtotta szolgáltatásokat, mint más helységekből érkező egyének.

2.2. Fogalmak

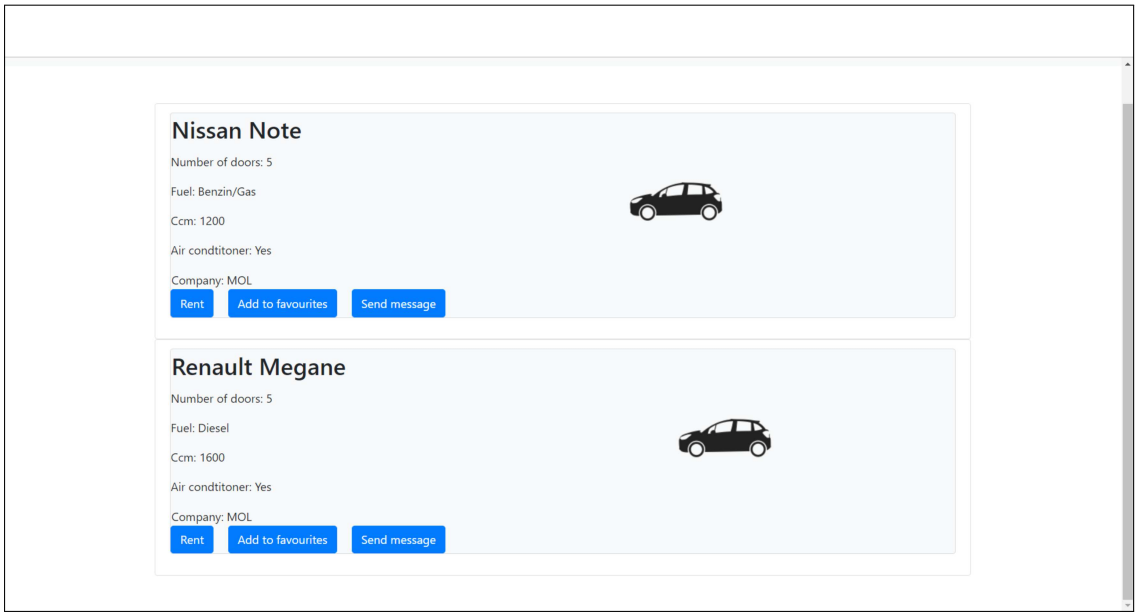
- Felhasználó - Egy felhasználó adatait tartalmazza: felhasználónév, kódolt jelszó, e-mail cím, jogosultság.
- Autó - egy autót reprezentál, rendelkezik gyártóval, típussal, ajtók számával, üzemanyag típussal, hengerűrtartalommal, rendelkezik-e klímával, illetve, hogy bérelve van-e. Minden autó típusnak van egy tulajdonosa, aki egy felhasználó.
- Motor - egy motort reprezentál, rendelkeznek gyártóval, típussal, hengerűrtartalommal illetve, hogy bérelve van-e. Minden motor típusnak van egy tulajdonosa, aki egy felhasználó.
- Bérlés - egy bérlés adatait tartalmazza, azaz bérlő személy, bérelt jármű, bérlés kezdetének ideje, bérlés végének ideje illetve, hogy ki van-e fizetve.
- Nyugta - egy bérlés kifizetése után keletkezik, tartalmazza a bérlő személy adatait, a bérelt jármű adatait, a bérlés kezdetét és a bérlés végét, valamint a költséget.
- Üzenet - egy üzenetet reprezentál, tartalmazza a küldő sorszámát, a fogadó sorszámát és az üzenet tartalmát.
- Kedvencek - egy felhasználó kedvencét reprezentálja, rendelkezik a felhasználó adataival és a kedvenc jármű adataival.

2.3. Guest felület

Guest jogosultságú felhasználó bejelentkezés után a kezdőoldalra lesz irányítva. Ez az oldal tartalmazza az éppen aktuális események információit. A fejlécben szereplő linkek segítségével tud az oldalak között navigálni a felhasználó. Az `autó` ikon az autók kilistázásához navigálja a felhasználót, a `motor` ikon pedig a motorok kilistázásához. Ezeken az oldalakon lesz lehetősége a bérlő személynek járművet bérelni, hozzáadni az adott eszközt a kedvenceihez, valamint üzenetet küldeni. A kezdőoldalon lehetőség van még a bejelentkezett felhasználó bérléseinek megtekintésére, a profil megtekintésére, valamint kijelentkezésre. A bérlések kilistázásánál lesz lehetősége a felhasználónak visszaszolgáltatni a bérelt járművet. Ezután ha szükségesnek érzi, értékelheti az adott cég által nyújtott szoláltatást. A következő lépésben a felhasználó kifizetheti a szolgáltatást. A felhasználó profilja tartalmazza a bejelentkezéshez használt információk egy részét, az üzeneteket, amiket küldött, az üzeneteket, amiket fogadott, valamint a kedvenc járműveit. A kezdőoldalon fel vannak tüntetve Budapest egyes legismertebb látványosságai, amelyekről rövid leírás olvasható. Ahhoz, hogy a felhasználó minél könnyebben tudjon tájékozódni, linkek vannak beállítva az egyes látványosságokhoz, melyek megjelenítik a térképen az objektum helyét.

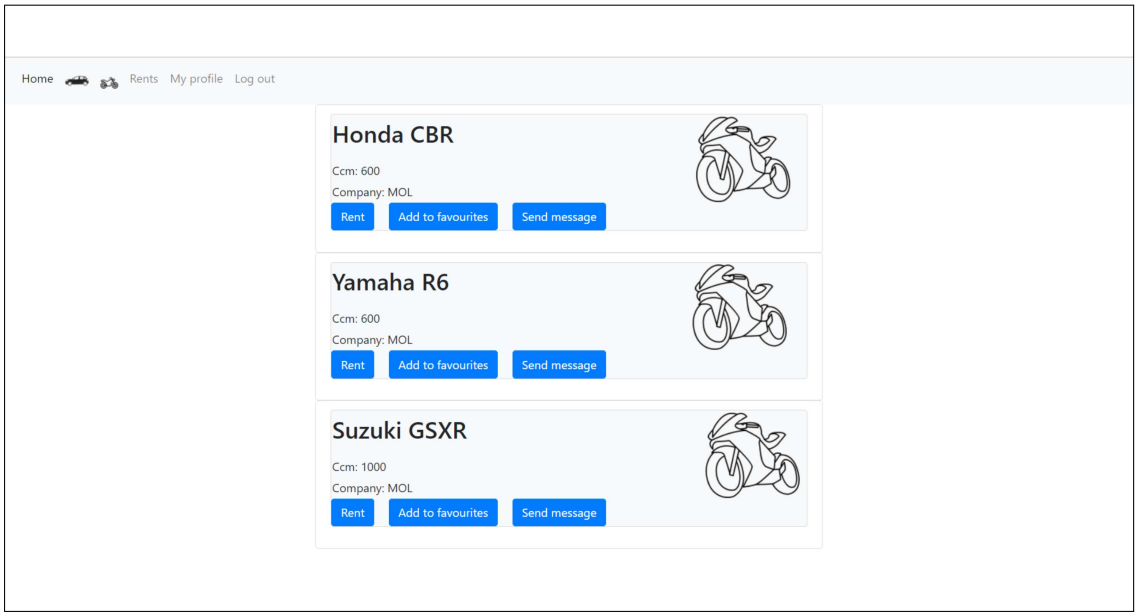
2.3.1. Járművek listázása, bérlése, kapcsolatfelvétel, felvétel a kedvencek közé

A járművek listázása az `autó` ikonra illetve a `motor` ikonra kattintva érhetők el. Ezeken az oldalakon jelennek meg az éppen bérelhető járművek. Az egyes járművek külön elválasztott szekciókban vannak megjelenítve. A szekciókban fel vannak tüntetve az eszközök jellemzői, motor esetében a gyártó neve, modell megnevezése, hengerűrtartalom és a cég, akihez az adott jármű tartozik, autó esetében a gyártó neve, modell megnevezése,fel van tüntetve, hogy hány ajtós gépkocsi, az üzemanyag típusa, hengerűrtartalom, hogy rendelkezik-e klímával az adott jármű és a cég, akihez az adott gépkocsi tartozik. Ezen az oldalon van lehetőség a bérlés megkezdésére, kapcsolatfelvételre a járműtulajdonossal, valamint a kedvencekhez való hozzáadáshoz.



2.1. ábra. Autók kilistázása

Ha a felhasználó bérelni kíván a felsorolt gépjárművek közül, akkor a `Rent` gombra kattintva megteheti azt. Ekkor a bérlés elmentésre kerül, és megtekinthető a `Rents` fejlécelemre kattintva. Ilyenkor a járműnél is történik változás. Mivel minden jármű rendelkezik egy "bérelve van-e" attribútummal, ezért minden bérlés esetén ezt az attribútumot meg kell változtatni. Ha egy járművet valaki kibérelt, akkor az addig nem jelenik meg a listázásban, amíg a bérlő személy vissza nem szolgáltatja az eszközt. Ha a bérlés megtörtént, az oldal újra az adott típusú járművek kilistázásához navigál. Nincsen korlátozva, hogy egy felhasználó hány eszközt bérelhet ki, tehát bármikor újra bérelhet járművet.



2.2. ábra. Motorok kilistázása

Ha a felhasználónak kérdése van egy adott eszközzel kapcsolatban, akkor azt felteheti a járműtulajdonosnak. Ezt a `Send message` gombra kattintva teheti meg. Ekkor egy új oldalra lesz navigálva a felhasználó, ahol beírhatja az üzenetét a címzettnek. Üzenetküldés esetén csak a tartalmat kell begépelnie a felhasználónak, amit el szeretne küldeni, a többi információt a program oldja meg. Amikor az üzenet küldésre kész, a felhasználó rákattint a `Send` gombra, amire az üzenet elküldésre kerül. Ezután a kezdőoldalra navigál a program. Ha a felhasználó megszeretné tekinteni az üzeneteit, akkor a profiljára kattintva megteheti azt. Itt a küldött és a fogadott üzenetek külön vannak kategorizálva. A fogadott üzeneteknél lehetőség van válasz megírására, ezt az `Answer` gombra kattintva megteheti a felhasználó, ami ugyanarra az oldalra navigálja őt, mint amikor üzenetet küldött. Több válasz is küldhető, ez az opció nem tűnik el az első válaszadás után. A harmadik opció, amit a felhasználó választhat ezen az oldalon a kedvencekhez való hozzáadás. Minden felhasználóhoz tartozik egy kedvencek lista, amiben szerepelnek a kedvencekhez hozzáadott járművek. Ha a felhasználónak megtetszik vagy a múltbéli tapasztalatai alapján megtetszett egy eszköz, akkor azt hozzáadhatja a kedvenceihez az `Add to favourites` gombra kattintva. Ekkor hozzáadódik a listához az adott jármű. Ha a hozzáadódás megtörtént, akkor egy zöld hátterű Success üzenet jelzi azt a képernyő jobb felső sarkában. A profil oldalon megtekinthetőek a kedvencek, ahol a motorok és az autók

szintén külön vannak választva, és mindkettőnek megjelennek egyes adatai, motor esetén a gyártó, modell, hengerűrtartalom, és a motor tulajdonosa, autó esetén a gyártó, modell, hengerűrtartalom, rendelkezik-e klímával és az autó tulajdonosa.

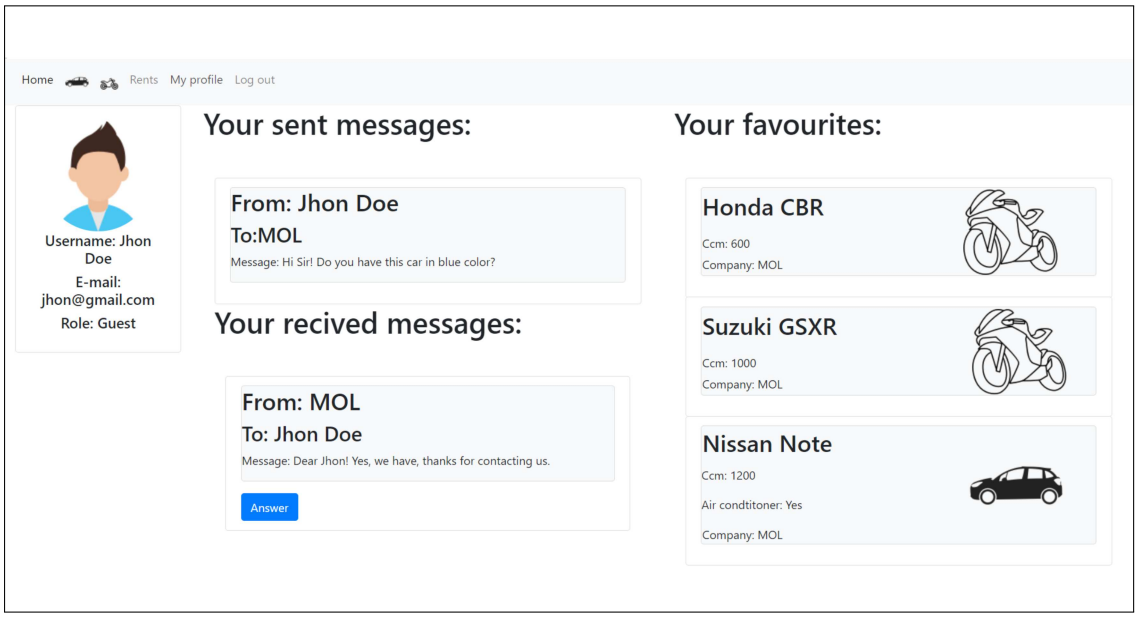
2.3.2. Bérlések

Minden Guest és Company jogosultságú felhasználó rendelkezik a bérléseiket kilistázó oldallal. Ezen a z oldalon jelennek meg a még aktív és a már befejezett bérlések. Minden bérlés esetén fel van tüntetve a bérelt jármű gyártója, modellje, fel van tüntetve, hogy ki bérelte ki, jelen esetben ez felhasználó, aki be van jelentkezve, szerepel, hogy kitől kölcsönözte az adott járművet, ezen kívül megjelenik a kölcsönzés kezdetének dátuma, és ha a bérlés már véget ért és a bérlő személy visszaszolgáltatta a járművet, akkor megjelenik a visszaadás dátuma is. Ha a felhasználó most szeretné befejezni a bérlést, akkor a `Return` gombra kattintva teheti meg azt. Ekkor frissül az oldal, és új opciók jelennek meg. Miután vége lett a kölcsönzésnek lehetőség nyílik a cég értékelésére, akiről a jármű származik. Ez opcionális, nem kötelező. Ha megtörtént az értékelés, akkor egy zöld hátterű Success üzenet jelzi ezt a képernyő jobb felső sarkában. A másik lehetőség az, hogy a felhasználó kifizeti a bérlés által keletkezett összeget. Ezt a `Pay` gombra kattintva teheti meg. Ekkor létrejön a nyugta, amit a program eltárol, és ez bármikor elérhetővé válik a cégnek, aki kiállította a nyugtát. A számla automatikusan letöltésre kerül, amit a felhasználó megtekinthet. Miután egy bérelt járművet visszaszolgáltatott a felhasználó, meg kell változtatni az eszköz "bérelve van-e" attribútumát, hogy újra megjelenjen a bérelhető eszközök listájában. Ezt a program a számla előállítása folyamán elvégzi. A fizetés után az alkalmazás a kezdőoldalra navigál.

2.3.3. Felhasználó profilja

Minden felhasználó rendelkezik egy profil oldallal, ahol a személyes információi és adatai láthatóak. Ezt a `My profile` navigációs sávbeli elemre kattintva teheti meg a felhasználó. Ekkor egy új oldalra kerül az alkalmazás használó. Ez a képernyő három oldalt jelenít meg. A felhasználó adatai, az üzenetek és a kedvencek mind külön oldalakként működnek. Van egy oldal, ami egybefogja ezeket, és biztosítja a megfe-

lelő elrendezésüket. Ezen az oldalon érhetők el számára a bejelentkezéskor használt információk, a jelszó kivételével. Minden felhasználó rendelkezik egy alapértelmezett Avatar képpel, felhasználónévvel, e-mail címmel, jogosultsággal (Guest, Company, Admin), valamint értékeléssel, ha már volt értékelve az adott felhasználó. Emellett ezen a képernyőn lesznek láthatóak az üzenetek, amiket a felhasználó küldött illetve fogadott. A harmadik része a képernyőnek pedig a kedvencek listáját jeleníti meg.



2.3. ábra. Felhasználó profilja

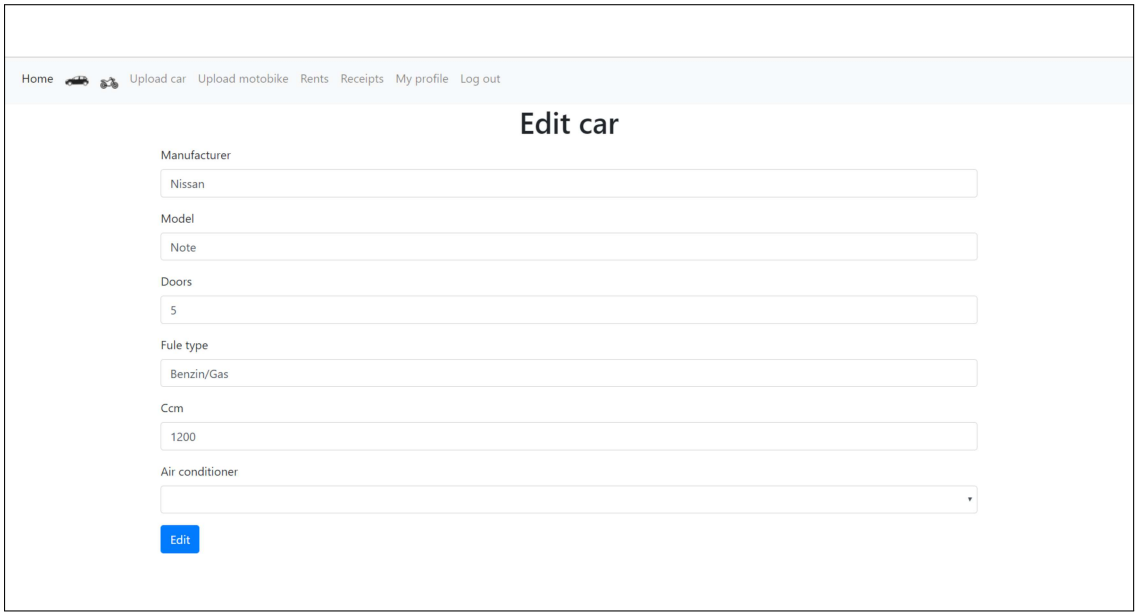
2.4. Company felület

Company jogosultságú felhasználó bejelentkezés után a kezdőoldalra lesz irányítva. Ez az oldal tartalmazza az éppen aktuális események információit. A fejlécben szereplő linkek segítségével tud az oldalak között navigálni a felhasználó. Az `autó` ikon az autók kilistázásához navigálja a felhasználót, a `motor` ikon pedig a motorok kilistázásához. Ezeken az oldalakon lesz lehetősége a bérlő személynek járművet bérelni, hozzáadni az adott eszközt a kedvenceihez, üzenetet küldeni, módosítani az adott járművet, valamint törölni azt. A kezdőoldalon lehetőség van még új autó illetve motor feltöltésére, a cégtől bérelt járművek kilistázására, a már véget ért kölcsönzések számláinak újra letöltésére, a profil megtekintésére, valamint kijelentkezésre. A bérlések kilistázásánál lesz lehetősége a felhasználónak visszaszolgáltatni a bérelt járművet. Ezután ha szükségesnek érzi, értékelheti az adott cég által nyúj-

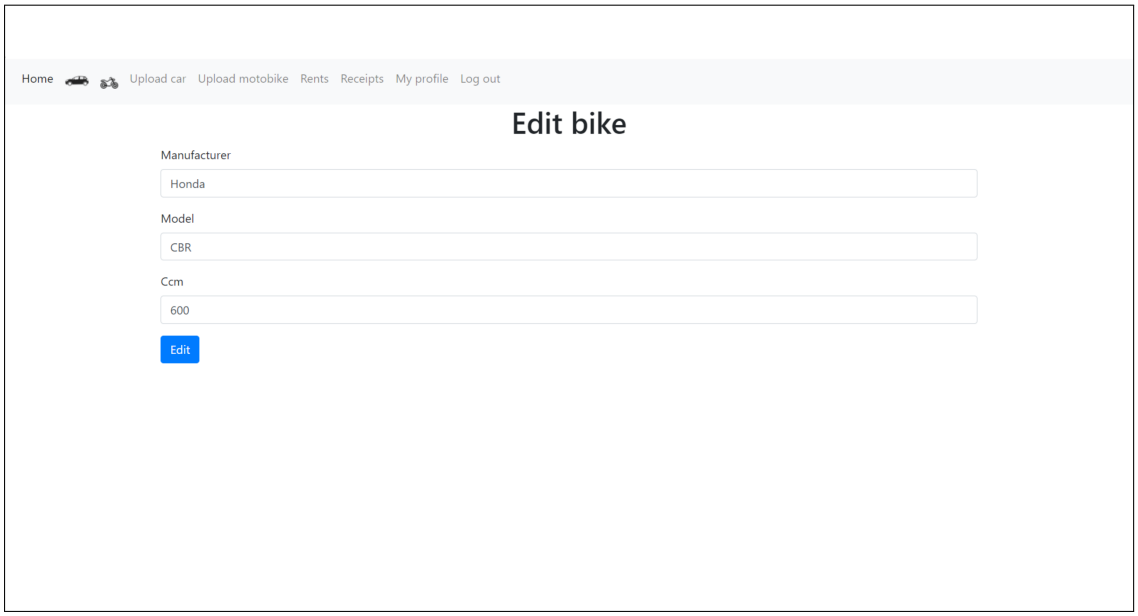
tott szoláltatást. A következő lépésben a felhasználó kifizetheti a szolgáltatást. A felhasználó profilja tartalmazza a bejelentkezéshez használt információk egy részét, az üzeneteket, amiket küldött, az üzeneteket, amiket fogadott, valamint a kedvenc járműveit.

2.4.1. Járművek módosítása és törlése

A járművek kilistázása annyiban tér el a Guest jogosultságú felhasználótól, hogy a cég módosíthatja a saját járműveinek az adatait illetve törölhet saját járművet. Módosításnál a felhasználó egy másik oldalra lesz navigálva, ahol megjelenik egy űrlap, ami a már tartalmazza a kiválasztott jármű adatait. A felhasználó ezen adatok alapján látja, hogy mit szeretne javítani. A javítás úgy működik, hogy a kiválasztott mezőbe beírja a felhasználó az új adatot és rányom az `Edit` gombra. Ennek hatására megváltoznak a jármű adatai, a felhasználó vissza lesz navigálva azokhoz a típusú eszközökhöz, amilyet éppen módosított. Egyszerre több adatott is lehet módosítani a járműnél, nem kell egyesével módosítgatni. Törlés esetén a felhasználó kiválasztja a listából a törölni kívánt elemet, és rákattint a `Delete` gombra. Ezután a jármű törlődik a rendszerből, a felhasználó pedig a kezdőoldalra lesz navigálva.



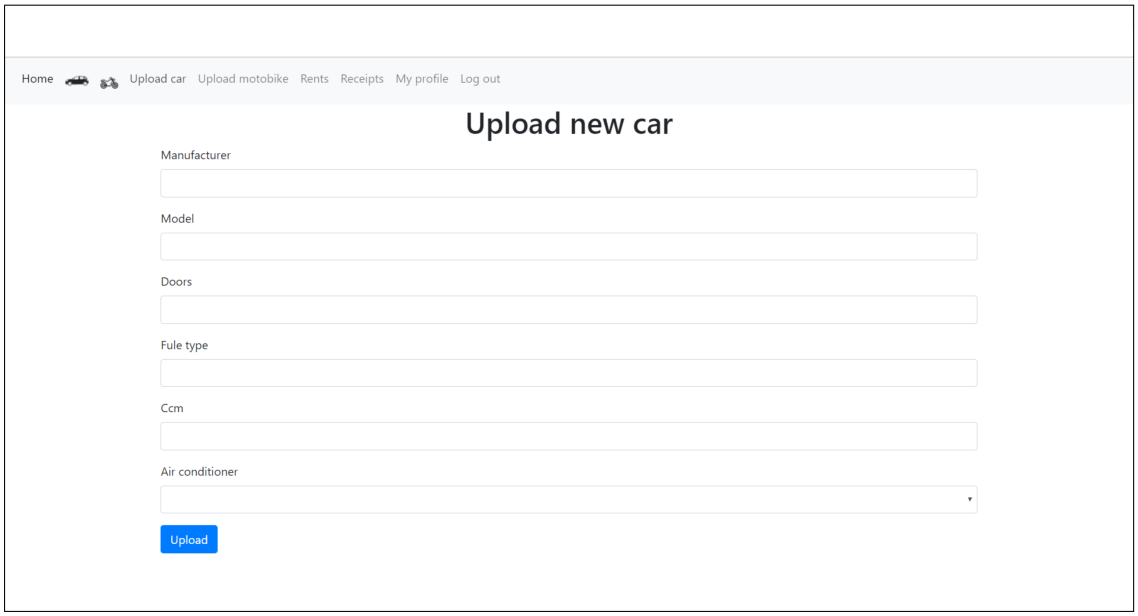
2.4. ábra. Autó módosítása



2.5. ábra. Motor módosítása

2.4.2. Autó feltöltése

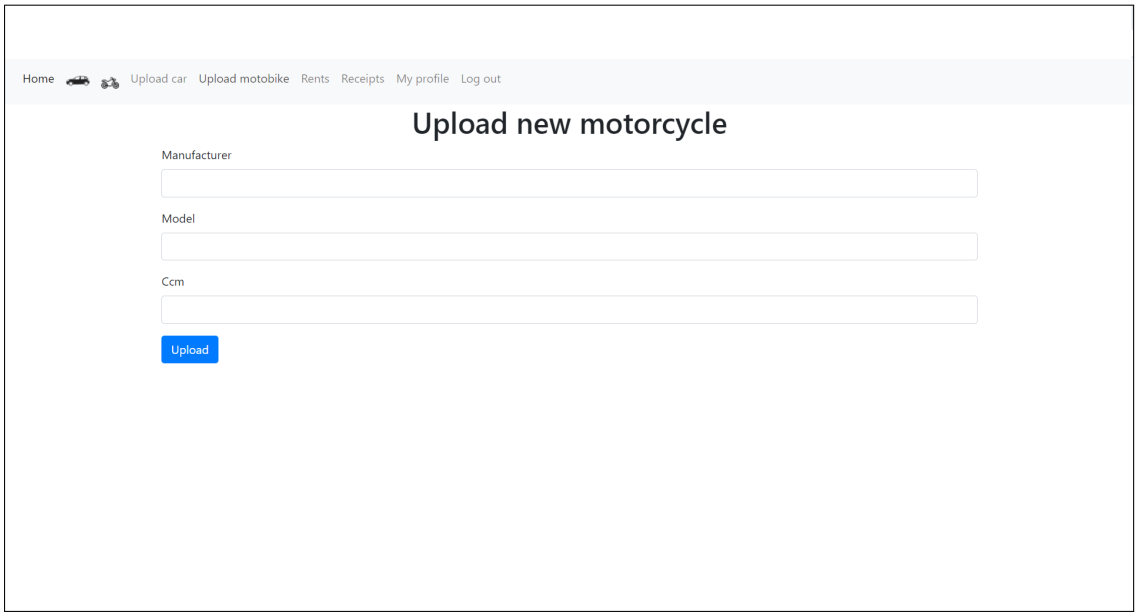
Ahhoz, hogy új járművek kerüljenek a rendszerbe cégekre van szükség, akik ezeket közzé teszik a felhasználók számára. A cégek miután bejelentkeztek és `Company` jogosultsággal bírnak, kiválaszthatják az `Upload car` fület. Ezen opció választása egy űrlaphoz navigálja őket. Itt kell feltüntetni a gépjármű egyes adatait. Autó esetén ezek a gyártó neve, modell, ajtók száma, üzemanyag típusa, hengerűrtartalom illetve, hogy rendelkezik-e klímával. Miután a felhasználó megadta ezen adatokat, a rendszer elvégzi a feltöltést. Ha a feltöltés megtörtént, akkor egy zöld hátterű Success üzenet jelzi azt a képernyő jobb felső sarkában. A felhasználó az autók kilistázásához lesz navigálva, ahol megtalálhatja a feltöltött járművet.



2.6. ábra. Autó feltöltése

2.4.3. Motor feltöltése

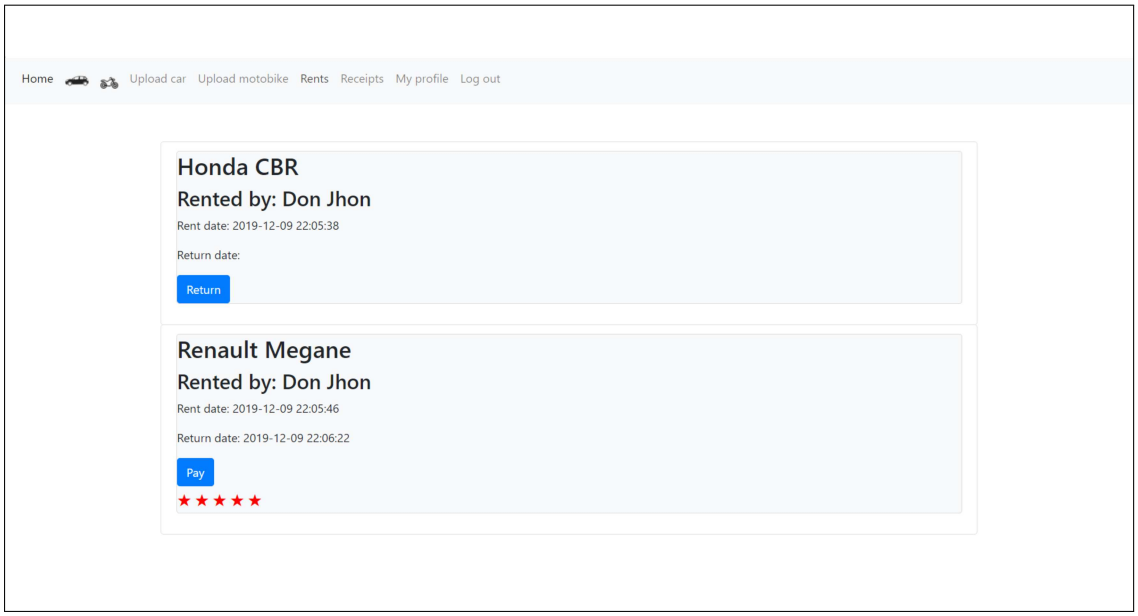
A cégek bejelentkezés után kiválasztják a `Upload motobike` fület. Ekkor egy űrlapra lesznek navigálva, ahol meg kell adni a motor egyes jellemzőit, melyek a gyártó neve, modell és a hengerűrtartalom. Miután a felhasználó ezeket megadta, rányomhat az `Upload` gombra. Ennek a gombnak a hatására bekerül a motor az adatbázisba, melyet egy zöld hátterű Success üzenet jelez a képernyő jobb felső sarkában. A felhasználó vissza lesz navigálva a motorok kilistázásához, ahol megtekintheti a feltöltött járművet.



2.7. ábra. Motor feltöltése

2.4.4. A cégektől bérelt járművek kilistázása

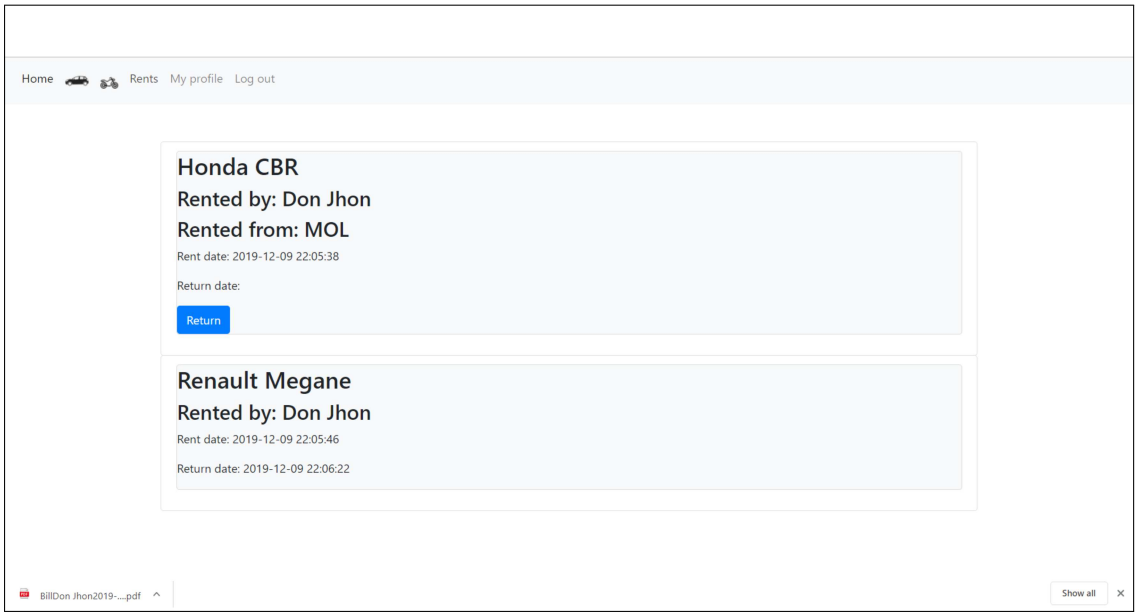
A Company jogosultsággal rendelkező egyéneknek lehetőségük van megtekinteni az általuk kölcsönbe adott gépjárműveket. Ezt a `Rents` fülre kattintva tehetik meg. Itt látható az összes jármű, amit az adott cégtől valaha béreltek illetve azok is, amik még bérelve vannak. Azok a kölcsönzések, melyek véget értek rendelkeznek visszaszolgáltatás dátummal. Ezeken a bérléseken semmilyen művelet nem hajtható végre. A még folyamatban lévő kölcsönzések esetén a cég visszaveheti az adott járművet. Ezt a `Return` gombra kattintva teheti meg. Ekkor a jármű ugyanúgy visszakerül a bérelhető eszközök közé, mintha a bérlő személy szolgáltatta volna vissza. Ez hasznos lehet, ha a kölcsönző felhasználó elfelejtette a jelszavát vagy felhasználónevét és nem tud bejelentkezni a rendszerbe. A cég miután visszavette a járművet előállíthatja a számlát is, azaz a kifizetést is megvalósíthatja. Ez ugyanazon esetekben hasznos, mit a kölcsönzés lezárása. Ha a bérlő személy képes bejelentkezni az alkalmazásba, akkor ő is elvégezheti ezeket a műveleteket. Ha a cég visszavette az adott eszközt, de a fizetést még nem indította el, akkor ezt a bérlő személy megteheti, és minden további funkció elérhető lesz számára.



2.8. ábra. A MOL cégtől bérelt járművek

2.4.5. Nyugta letöltése

Miután egy eszköz bérlése véget ér lehetőség nyílik a kölcsönzés kifizetésére. Ezt a bérlő személy nem köteles azonnal megtenni. A rendszer megjegyzi, hogy mely bérlések lettek kifizetve. A bérlő személy csak akkor kapja meg a számlát, ha kifizette a kölcsönzést. Ez egyszer történhet meg. Viszont mi van akkor, ha a nyugta elveszik, de a bérlőnek szüksége van a nyugtára? A megoldás abban rejlik, hogy az összes számla el van tárolva az adatbázisban. Ez alapján a cég bármikor képes új nyugta nyomtatására az eredeti bérlés adataival. Ezt a cégek a `Receipts` fülön tehetik meg. Itt jelennek meg a már véget ért kölcsönzések, amelyekről nyugta készíthető. A listában minden kölcsönzés elkülönítve szerepel, és rendelkezik a bérlés ismertető adataival, a bérelt járművel és típusával, a bérlő személy nevével és az összeggel. Ezek alatt szerepel a `Create receipt` gomb melyre rákattintva a cég kiállíthatja a számlát, ami automatikusan letöltésre kerül. Ez a művelet akárhányszor megismételhető, nincs korlátozva a számla létrehozások száma.



2.9. ábra. Nyugta letöltése

2.4.6. Felhasználó profilja

Minden felhasználó rendelkezik egy profil oldallal, ahol a személyes információi és adatai láthatóak. Ezt a `My profile` navigációs sávbeli elemre kattintva teheti meg a felhasználó. Ekkor egy új oldalra kerül az alkalmazás használó. Ez a képernyő három oldalt jelenít meg. A felhasználó adatai, az üzenetek és a kedvencek mind külön oldalakként működnek. Van egy oldal, ami egybefogja ezeket, és biztosítja a megfelelő elrendezésüket. Ezen az oldalon érhetők el számára a bejelentkezéskor használt információk, a jelszó kivételével. Minden felhasználó rendelkezik egy alapértelmezett Avatar képpel, felhasználónévvel, e-mail címmel, jogosultsággal (Guest, Company, Admin), valamint értékeléssel, ha már volt értékelve az adott felhasználó. Emellett ezen a képernyőn lesznek láthatóak az üzenetek, amiket a felhasználó küldött illetve fogadott. A harmadik része a képernyőnek pedig a kedvencek listáját jeleníti meg.

2.5. Admin felület

2.5.1. Járművek listázása

Az admin jogosultsággal rendelkező felhasználókból nem kell, hogy sok legyen. Ezek a felhasználók lesznek, akik karbantartják a rendszert. Nekik lesz a feladatuk

a felhasználók törlése, valamint a hírdetések törlése, ha az adott cég ezt nem hajtja végre. Moderátor szerepet töltenek be az adminok. Bejelentkezés után a kezdőképernyőre navigál a program. Ez a képernyő ugyanaz, mint a Guest és Company jogosultságú felhasználóknál. Az `autó` ikonra kattintva ugyanúgy kilistázásra kerülnek a gépkocsik, mint eddig. A változás az opciókban van. Mivel az adminok karbantartók ezért ők nem fognak járműveket bérelni, nem fogják hozzáadni a kedvenceikhez őket illetve módosítani sem fogják őket. Nekik van lehetőségük üzenetküldésre és törlésre. Az üzenetküldés lehet egy figyelmeztető üzenet, ha a cég nem megfelelő tartalmat tölt fel, lehet javaslat, ha a cég hibázott a feltöltéskor, vagy bármi egyéb. A törlés opció kiválasztásával a jármű végérvényesen törlődik a rendszerből. Ez például hasznos lehet, ha a cég nem megfelelő tartalmat töltött fel, sértő vagy illetlen szöveggel, ekkor az admin felveheti a kapcsolatot vele, hogy javítsa a hirdetést, vagy azonnal törölheti is azt.

2.5.2. Kölcsönzések listázása

Az Admin jogosultsággal rendelkező felhasználó kilistázhatja a kölcsönzéseket. Ez annyiban tér el a Company jogosultsággal rendelkező felhasználó kilistázásától, hogy az Admin felhasználó mindenkinek a kölcsönzését látja, nem csak egy adott cégét. Ugyanúgy lehetősége van a felhasználónak átvenni a bérelt eszközt, valamint végrehajtani a kifizetést, mint a Company jogosultságú felhasználónak.

2.5.3. Nyugta letöltése

Adminként minden bérléshez hozzáfér a felhasználó. Így minden bérlésre képes nyugtát előállítani, nem csak adott cégekhez tartozó nyugtákat. Hasonlóan, mint a Company jogosultsággal bejelentkezett felhasználóknál, itt is korlátlan alkalommal előállítható a számla az eredeti adatokkal, melyek az adatbázisban vannak eltárolva.

2.5.4. Felhasználók törlése

Az admin felhasználóknak két nagy erejük van, bárkinek a hirdetését törölhetik, valamint felhasználókat törölhetnek a rendszerből. Most az utóbbiról lesz szó. A `Users` fülre kattintva a navigációs sávban kilistázásra kerülnek a rendszerbe be-

regisztrált felhasználók. Minden felhasználó esetében fel van tüntetve a felhasználóneve, emailcíme és a jogosultsága. Ha törölni kell felhasználót, azt a `Delete` gombra kattintva teheti meg az admin. Ekkor a felhasználó törlődik a listából. Azok a felhasználók, akik a kitörölt felhasználónak írtak a múltban üzenetet, azoknál úgy jelenik meg az üzenet a profiljukon, hogy törölt felhasználónak volt címezve, de láthatóak maradnak. A törölt felhasználónak törlődnek a járművei is, azok a bérlések, amelyeket ő adott ki illetve a számlái is.

2.5.5. Felhasználó profilja

Az Admin jogosultságú felhasználók is megtekinthetik saját profiljukat. Ezt a `My profile` navigációs sávbeli elemre kattintva teheti meg a felhasználó. Ekkor egy új oldalra kerül az alkalmazás használó. Ez a képernyő két oldalt jelenít meg. A felhasználó adatai és az üzenetek. Van egy oldal, ami egybefogja ezeket, és biztosítja a megfelelő elrendezésüket. Ezen az oldalon érhetők el számára a bejelentkezéskor használt információk a jelszó kivételével. Emellett ezen a képernyőn lesznek láthatóak az üzenetek, amiket a felhasználó küldött illetve fogadott.

3. fejezet

Fejlesztői dokumentáció

3.1. A probléma specifikációja

Az alkalmazás egy város közlekedési eszközeinek a kibérlésére lett elkészítve. A rendszer célja, hogy mindenkinek, elsősorban a turistáknak, egyszerű, gyors és könnyű hozzáférést biztosítson a kibérelhető járművekhez, azaz sorban állás nélkül a felhasználók bérelhessenek eszközöket. A rendszer backendje Java nyelven lett megírva, Spring keretrendszer használatával, a frontend pedig TypeScript nyelven, Angular segítségével. Az alkalmazásnak két fő funkciója van: egyrészt lehetővé teszi a közlekedési eszközök feltöltését a rendszerbe, másrészt pedig kölcsönzés folyamatát valósítja meg, valamint annak következményeit. Az adatbevitel űrlapok segítségével történik. Az alkalmazás követi a *model-view-controller* (modell-nézet-vezérlés) tervezési mintát, saját modell, nézet és vezérlési fájlokkal rendelkezik.

3.2. Felhasznált eszközök

- A Java [1] egy általános célú, objektumorientált programozási nyelv. A Java alkalmazásokat jellemzően bájtkód formátumra alakítják, de közvetlenül natív (gépi) kód is készíthető Java forráskódból. A bájtkód futtatása a Java virtuális géppel történik

- A Spring [2] egy nyílt forráskódú, inversion of controlt megvalósító Java alkalmazás keretrendszer. A Spring keretrendszer magját képező szolgáltatásokat főként Java alkalmazás fejlesztésére használják.

- A TypeScript egy objektum-orientált statikusan típusos script nyelv, amit a Microsoft készített.
- Az Angular [3] egy TypeScript alapú nyílt forráskódú webes alkalmazás keretrendszer, melyet a Google fejlesztett. Ez a keretrendszer sokkal egyszerűbbé teszi a frontend fejlesztését.
- A Bootstrap [4] egy nyílt forráskódú frontend keretrendszer, ami reszponzív weboldalak létrehozását segíti elő.

3.3. Az alkalmazás felépítése

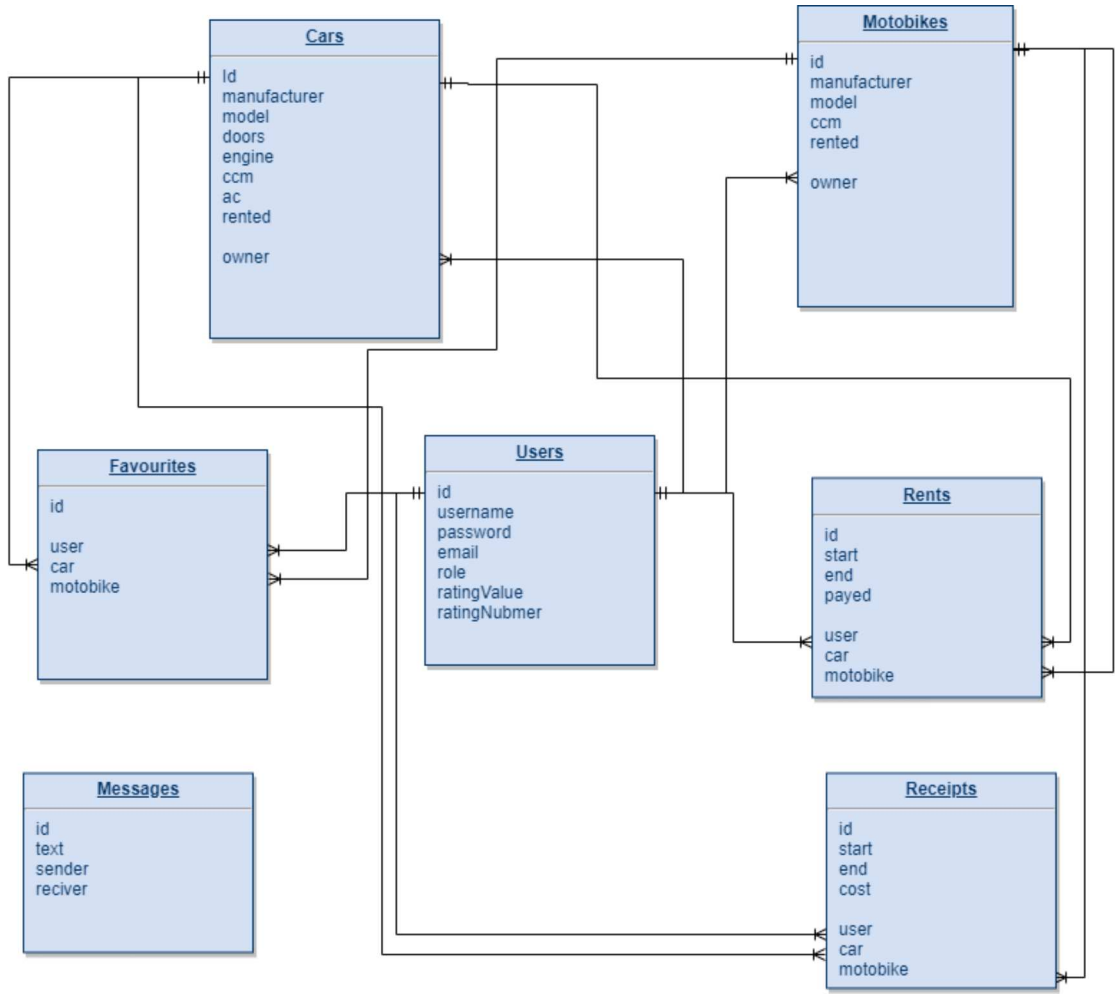
Az alkalmazás két részből áll: backend, frontend. A backend végzi a szerveroldali számításokat, itt találhatóak a *modell eszközök* és a *controllerek*. A frontend egy külön könyvtárba van szervezte melynek neve `frontend`. Itt valósul meg a nézet.

3.4. Model-view-controller

A model-view-controller, vagy magyarul modell-nézet-vezérlő egy tervezési minta, amit gyakran alkalmaznak webes alkalmazások készítéséhez. Mint az a nevéből is látszik, az alkalmazást három különálló részre bontja. A modell réteg felelős az adatok tárolásáért és az adatokon értelmezett műveletekért. A nézet a felhasználói felületet foglalja magába. A vezérlő pedig az utóbbi kettő között teremt kapcsolatot; a felhasználó inputját továbbítja a modell felé, illetve a modell által tárolt adatokhoz hozzáférést biztosít.[5]

3.5. Modell réteg

A `model` mappában szerepelnek az alkalmazásban használt táblák.



3.1. ábra. Az adatbázis séma egyed-kapcsolat diagramja

3.5.1. A modul táblái

Az alkalmazáshoz hét táblát hoztam létre. Mindegyik táblában az azonosító az elsődleges kulcs. Minden táblát egyesével, külön-külön részletezek táblázatba foglalva.

Users

Ez a tábla tartalmazza a felhasználók adatait. Aki regisztrál az alkalmazásba ebben a táblában megtalálható, és a későbbiekben ennek a táblának a segítségével lesznek megoldhatóak egyes műveletek. A jogosultságok felsorolóval vannak megadva, melyek `Role` típusúak.

Oszlopok	
Név és típus	**Magyarázat**
id: int	azonosító
username: String	felhasználónév
password: String	a felhasználó kódolt jelszava
email: String	a felhasználó e-mail címe
role : Role	jogosultság
ratingValue: int	az értékelések összege
ratingNumber: int	az értékelések száma
Idegen kulcsok	
Név	**Magyarázat**
cars	a felhasználóhoz tartozó autók, a `Cars` táblára hivatkozik
motobikes	a felhasználóhoz tartozó motorok, a `Motobikes` táblára hivatkozik
rent	a felhasználóhoz tartozó kölcsönzések, a `Rents` táblára hivatkozik
favourites	a felhasználó kedvenc járművei, a `Favourites` táblára hivatkozik
receipt	a felhasználó nyugtáit tartalmazza, a `Receipts` táblára hivatkozik

3.1. táblázat. Users tábla

Cars

Ez a tábla egy autót reprezentál. A `Cars` és a `Rents` táblák között, a `Cars` és a `Favourites` között valamint a `Cars` és a `Receipts` táblák között egy-sok kapcsolat van. A `Cars` és a `Users` táblák között sok-egy kapcsolat van.

Oszlopok	
Név és típus	**Magyarázat**
id: int	azonosító
manufacturer: String	gyártó
model: String	típus
doors : int	az ajtók száma
engine: string	üzemanyag típusa
ccm: int	hengerűrtartalom
ac: Boolean	rendelkezik-e klímával
rented: Boolean	ki van-e bérelve
Idegen kulcsok	
Név	**Magyarázat**
owner	a `Users` táblára egy adott sorára hivatkozik az `ownerID` alapján
rent	a `Rents` tábla egy adott sorára hivatkozik
favourites	a `Favourites` tábla egy adott sorára hivatkozik
receipts	a `Receipts` tábla egy adott sorára hivatkozik

3.2. táblázat. Cars tábla

Motobikes

Ez a tábla egy motort reprezentál. A `Motobikes` és a `Rents` táblák között, a `Motobikes` és a `Favourites` között valamint a `Motobikes` és a `Receipts` táblák között egy-sok kapcsolat van. A `Motobikes` és a `Users` táblák között sok-egy kapcsolat van.

Oszlopok	
Név és típus	**Magyarázat**
id: int	azonosító
manufacturer: String	gyártó
model: String	típus
ccm: int	hengerűrtartalom
rented: Boolean	ki van-e bérelve
Idegen kulcsok	
Név	**Magyarázat**
owner	a `Users` tábla egy adott sorára hivatkozik az `ownerID` alapján
rent	a `Rents` tábla egy adott sorára hivatkozik
favourites	a `Favourites` tábla egy adott sorára hivatkozik
receipt	a `Receipts` tábla egy adott sorára hivatkozik

3.3. táblázat. Motobikes tábla

Rents

Ez a tábla a kölcsönzések nyilvántartására szolgál. A `Rents` és a `Users`, a `Rents` és a `Cars` valamint a `Rents` és a `Motobikes` táblák sok-egy kapcsolatban vannak egymással.

Oszlopok	
Név és típus	**Magyarázat**
id: int	azonosító
start: Date	a kölcsönzés kezdete
end: Date	kölcsönzés vége
payed : Boolean	fizetve van-e
Idegen kulcsok	
Név	**Magyarázat**
user	a `Users` tábla egy adott sorára hivatkozik az `ownerID` alapján
car	a `Cars` tábla egy adott sorára hivatkozik
motobike	a `Motobikes` tábla egy adott sorára hivatkozik

3.4. táblázat. Rents tábla

Favourites

Ez a tábla tartalmazza a felhasználók kedvenceit. A `Faourites` és a `Users`, a `Favourites` és a `Cars` valamint a `Favourites` és a `Motobikes` táblák sok-egy kapcsolatban vannak egymással.

Oszlopok	
Név és típus	**Magyarázat**
id: int	azonosító
Idegen kulcsok	
Név	**Magyarázat**
user	a `Users` tábla egy adott sorára hivatkozik az `ownerID` alapján
car	a `Cars` tábla egy adott sorára hivatkozik
motobike	a `Motobikes` tábla egy adott sorára hivatkozik

3.5. táblázat. Favourites tábla

Messages

Ez a tábla tartalmazza az üzeneteket az egyes felhasználók között.

Oszlopok	
Név és típus	**Magyarázat**
id: int	azonosító
text: String	az üzenet szövege
sender: int	a küldő személy ID-ja
reciver: int	a fogadó ID-ja

3.6. táblázat. Messages tábla

Receipts

Ez a tábla tartalmazza a nyugtákat a kölcsönzésekről. A `Receipts` és a `Users`, a `Receipts` és a `Cars` valamint a `Reecipts` és a `Motobikes` táblák sok-egy kapcsolatban vannak egymással.

Oszlopok	
Név és típus	**Magyarázat**
id: int	azonosító
start: Date	a kölcsönzés kezdete
end: int	a kölcsönzés vége
cost: int	a fizetendő összeg
Idegen kulcsok	
Név	**Magyarázat**
user	a `Users` tábla egy adott sorára hivatkozik az `ownerID` alapján akié a nyugta
car	a `Cars` tábla egy adott sorára hivatkozik, amely járművet bérelték
motobike	a `Motobikes` tábla egy adott sorára hivatkozik, amely motort bérelték

3.7. táblázat. Receipts tábla

3.6. Vezérlő réteg

Az alkalmazás minden táblája rendelkezik egy vezérlővel. Ezek a vezérlők adják meg a választ a kapott kérésekre.

3.6.1. UsersController

Ez a réteg a felhasználók kezelésére szolgál. Rendelkezik egy `UserRepository` objektummal, ami egy interface. Ebben az interfaceban a kereséshez szükséges függvények vannak deklarálva(findALL(), findByUsername(), findById()). A controllerben deklarálva vannak a következő függvények:

- getUsers(), amely kilistázza az összes beregisztrált felhasználót.
- getUser(int id), amely a kapott azonosító alapján kikeresi a megfelelő felhasználót, és ha talál ilyet, akkor visszatér vele.
- register(User user), amely a kapott felhasználót beregisztrálja a rendszerbe.
- login(), amely az azonosított felhasználót belépteti a rendszerbe.
- addRate(User user, int id), amely a megadott ID-val rendelkező felhasználónak frissíti az értékelését.
- deleteUser(int id), ezt csak admin jogosultsággal rendelkező felhasználók használhatják, és ezzel kitörölhetnek egy adott felhasználót.

3.6.2. CarsController

Ezen vezérlőréteg az autók kezelését segíti. Rendelkezik egy `UserRepository` és egy `CarsRepository` objektummal, ami a egy interface. Ebben szerepelnek a járművek eléréséhez szükséges függvények deklarációi(findById(), findByOwnerId(), findAllByOwnerId(), findAll()). A controllerben deklarálva vannak a következő függvények:

- getCars(), amely kilistázza az összes autót, ami a rendszerben szerepel.
- getCar(int id), amely a kapott azonosító alapján kikeresi a megfelelő járművet, és ha talál ilyet, akkor visszatér vele.
- getCarsByUser(int owner), amely a megadott felhasználó összes járművét kilistázza.
- addCar(Car car, int owner), amely a megadott felhasználóhoz hozzárendel egy új autót, cég használhatja.
- deleteCar(int id), ezt csak admin vagy cég jogosultsággal rendelkező felhasználók használhatják, és ezzel kitörölhetnek egy adott autót.
- deleteCarByOwner(int id), amely a megadott felhasználó azonosítója alapján töröl ki egy járművet, csak admin jogosultsággal használható.

- putCar(Car car, int id), amely a megadott azonosítóval rendelkező autót felülírja a paraméterben átadottal. Ezt csak cégek végezhetik el a saját járműveiken.

3.6.3. BikesController

Ezen vezérlőréteg motorok kezelését segíti. Rendelkezik egy `UserRepository` és egy `BikesRepository` objektummal, ami a egy interface. Ebben szerepelnek a járművek eléréséhez szükséges függvények deklarációi(findById(), findByOwnerId(), findAll(), findAllByOwnerId()). A controllerben deklarálva vannak a következő függvények:

- getBikes(), amely kilistázza az összes motort, ami a rendszerben szerepel.
- getBikes(int id), amely a kapott azonosító alapján kikeresi a megfelelő gépjárművet, és ha talál ilyet, akkor visszatér vele.
- getBikesByUser(int owner), amely a megadott felhasználó összes járművét kilistázza.
- addBike(Motobike motobike, int owner), amely a megadott felhasználóhoz hozzárendel egy új motort, cég használhatja.
- deleteBike(int id), ezt csak admin vagy cég jogosultsággal rendelkező felhasználók használhatják, és ezzel kitörölhetnek egy adott motort.
- deleteBikeByOwner(int id), amely a megadott felhasználó azonosítója alapján töröl ki egy járművet, csak admin jogosultsággal használható.
- putBike(Motobike motobike, int id), amely a megadott azonosítóval rendelkező motort felülírja a paraméterben átadottal. Ezt csak cégek végezhetik el a saját járműveiken.

3.6.4. RentController

Ezen vezérlőréteg a kölcsönzések használatában segít. Rendelkezik egy `RentsRepository`, `UserRepository`, `CarsRepository` és egy `BikesRepository` objektummal, ami a egy interface. A `RentsRepositoryban` szerepelnek a kölcsönzések

eléréséhez szükséges függvények deklarációi(findByUserId(), findAll()). A controllerben deklarálva vannak a következő függvények:

- getRents(), amely kilistázza az összes kölcsönzést, ami a rendszerben szerepel.
- getRent(int id), amely a kapott azonosító alapján kikeresi a megfelelő kölcsönzést, és ha talál ilyet, akkor visszatér vele.
- getRentsByUser(int user), amely a megadott felhasználó összes kölcsönzését kilistázza.
- addRent(Rent rent, int useID, int carID, int motobikeID), amely a megadott felhasználóhoz hozzárendel egy új kölcsönzést és beállítja a megfelelő járművet.
- deleteRent(int id), amellyel egy kölcsönzés törölhető ki.
- returnRentedItem(Rent rent, int id), amely a megadott azonosítóval rendelkező kölcsönzést megváltoztatja úgy, hogy a kölcsönzés befejeződött, a jármű vissza lett szolgáltatva.
- payRentedItem(Rent rent, int id), amely hozzáveszi a megadott kölcsönzéshez az összeget, amit ki kell fizetni.

3.6.5. ReceiptController

Ezen vezérlőréteg a nyugták elérésében segít. Rendelkezik egy `ReceiptRepository`, `UserRepository`, `CarsRepository` és egy `BikesRepository` objektummal, ami egy interface. A `ReceiptRepositoryban` szerepelnek a nyugták eléréséhez szükséges függvények deklarációi(findByUserId(), findAll()). A controllerben deklarálva vannak a következő függvények:

- getReceipts(), amely kilistázza az összes számlát, ami a rendszerben szerepel.
- getReceipt(int id), amely a kapott azonosító alapján kikeresi a megfelelő nyugtát, és ha talál ilyet, akkor visszatér vele.
- getReceiptsByUser(int user), amely a megadott felhasználó összes számláját kilistázza.

- addReceipt(Receipt receipt, int useID, int carID, int motobikeID), amely a megadott felhasználóhoz hozzárendel egy új nyugtát és beállítja a megfelelő járművet.

3.6.6. FavouritesController

Ezen vezérlőréteg a kedvencek elérésében segít. Rendelkezik egy `FavouritesRepository`, `UserRepository`, `CarsRepository` és egy `BikesRepository` objektummal, ami a egy interface. A `FavouritesRepositoryban` szerepelnek a nyugták eléréséhez szükséges függvények deklarációi(findAllByUserId(),findByUserId(), findAll()). A controllerben deklarálva vannak a következő függvények:

- getFavourites(), amely kilistázza a felhasználók összes kedvenceit, ami a rendszerben szerepel.
- getFavourite(int id), amely a kapott azonosító alapján kikeresi a megfelelő kedvencet, és ha talál ilyet, akkor visszatér vele.
- getFavouriteByUser(int user), amely a megadott felhasználó összes kedvenceit kilistázza.
- addFavourite(Favourites fac, int useID, int carID, int motobikeID), amely a megadott felhasználóhoz hozzárendel egy új kedvencet és beállítja a megfelelő járművet.
- deleteFavouritesByOwner(int id), ami a megadott azonosítóval rendelkező felhasználónak törli a kedvencét.

3.6.7. MessagesController

Ezen vezérlőréteg az üzenetek elérésében segít. Rendelkezik egy `MessagesRepository` és egy `UserRepository` objektummal, ami a egy interface. A `MessagesRepositoryban` szerepelnek az üzenetek eléréséhez szükséges függvények deklarációi(findBySender(),findAllBySender(), findAllByReciver()). A controllerben deklarálva vannak a következő függvények:

- getMessages(), amely kilistázza a felhasználók összes üzeneteit, ami a rendszerben szerepel.
- getMessage(int id), amely a kapott azonosító alapján kikeresi a megfelelő üzenetet, és ha talál ilyet, akkor visszatér vele.
- getMessagesBySender(int sender), amely a megadott küldő alapján kikeresi az összes üzenetet, amit ő küldött.
- getMessagesByReciver(int reciver), amely a megadott fogadó alapján kikeresi az összes üzenetet, ami neki volt címezve.
- addMessage(Message message, int sender, int reciver), amely a megadott felhasználóhoz hozzárendel egy új üzenetet és beállítja az üzenet fogadóját.
- deleteFavouritesByOwner(int id), ami a megadott azonosítóval rendelkező felhasználónak törli a kedvencét.

3.7. Nézet réteg

A megjelenítéshez Bootstrap-et használtam, hogy gazdagítsa a felhasználói élményt. Több helyen speciális eszközöket is alkalmaztam, mint például az értékelésnél, ahol csillagok segítségével lehet megadni az elégedettségi szintet. Ehhez bővítményt használtam, ami az `ng-starrating` [6] névre hallgat. A másik eszköz, amit igénybe vettem a Success üzenetek, amik megjelennek az egyes oldalakon valamilyen akció végrehajtása után. Ez a bővítmény az `ngx-toastr` [7]. Ez annyiban gazdagítja a felhasználói élményt, hogy az akció sikeres végrehajtása után egy zöld mező jelenik meg a képernyő jobb felső sarkában, ami tartalmazza az akció leírását, és hogy sikeres volt. A nézet réteget öt fő kategóriába sorolnám, mivel számtalan oldallal rendelkezik. Ezek a kategóriák foglalják egybe az egyes csoportokat, amelyek valamilyen közös jellemzővel rendelkeznek. A nézet réteg fő részei:

- Autók
- Motorok
- Felhasználói profil

- Alapkomponensek
- Egyéb

3.7.1. Autók

Az autók kilistázásának megjelenítéséért a `car.component.html` felelős. Ez az oldal egy listát jelenít meg. A lista elemei tartalmazzák a járműveket. Ezen belül minden jármű esetén egy táblázatban jeleníti meg annak adatait. A táblázat két oszlopos, de a második oszlop sorai össze vannak vonva egyetlen sorrá, ebben helyezkedik el a jármű képe. Az első oszlop sorai fontossági sorrendben tartalmazzák az adatokat az autóról. A táblázat után pedig szerepelnek a gombok. Alapvetően hat gomb van feltüntetve, de mind a hat sosem lesz látható egy felhasználó számára. Attól függően, hogy ki listázza ki a járműveket, admin esetén kettő, cég esetén öt, vendégek esetén három gomb lesz látható.

Az autók feltöltésének megjelenítéséért a `uploadcar.component.html` a felelős. Ezen az oldalon egy űrlap található. Az adatokat egy `FormGroup`-ba kell beírni. Autók esetén a megadandó adatok: gyártó, modell, ajtók száma, üzemanyag típusa, hengerűrtartalom valamint a klíma. Az űrlap utolsó eleme egy gomb, ami a feltöltés elindítását végzi. A járművek módosításakor az `edit-car.component.html` jelenik meg. Ez az oldal egy az egyben megegyezik a feltöltés oldallal, itt is űrlap van, aminek kitöltésével lesz módosítva egy jármű.

3.7.2. Motorok

A motorok kilistázásának megjelenítéséért a `morotcycle.component.html` felel. Ez az oldal egy listát jelenít meg. A lista elemei tartalmazzák a járműveket. Ezen belül minden jármű esetén egy táblázatban jeleníti meg annak adatait. A táblázat két oszlopos, de a második oszlop sorai össze vannak vonva egyetlen sorrá, ebben helyezkedik el a jármű képe. Az első oszlop sorai fontossági sorrendben tartalmazzák az adatokat a motorról. A táblázat után pedig szerepelnek a gombok. Alapvetően hat gomb van feltüntetve, de mind a hat sosem lesz látható egy felhasználó számára. Attól függően, hogy ki listázza ki a járműveket, admin esetén kettő, cég esetén öt, vendégek esetén három gomb lesz látható.

A motorok feltöltésének megjelenítéséért a `uploadmotobike.component.html` a felelős. Ezen az oldalon egy űrlap található. Az adatokat egy `FormGroup`-ba kell beírni. Motorok esetén a megadandó adatok: gyártó, modell valamint hengerűrtartalom. Az űrlap utolsó eleme egy gomb, ami a feltöltés elindítását végzi. A járművek módosításakor az `edit-motobike.component.html` jelenik meg. Ez az oldal egy az egyben megegyezik a feltöltés oldallal, itt is űrlap van, aminek kitöltésével lesz módosítva egy jármű.

3.7.3. Felhasználói profilkörnyezet

A felhasználó profilját a `profile-panel.component.html` fogja össze. Ez a komponens szolgál alapul a profil oldal megjelenítésekor. Ezen az oldalon vannak a többi oldalak, amik tartalmazzák az adatokat a felhasználóról. Három komponens megjelenítését végzi: profile-data.component, profile-messages.component, profile-favourites.component. A `profile-data.component.html` tartalmazza a felhasználó profiljának adatait, amit a bejelentkezéshez használt. Ez az oldal akkor jelenik csak meg, ha van bejelentkezett felhasználó. A megjelenítendő adatok pedig az Avatar kép, felhasználónév, e-mail cím, jogosultságának köre illetve az értékelése, de csak akkor, ha már valaki értékelte. A `profile-messages.component.html` tartalmazza at adott felhasználó által küldött illetve fogadott üzeneteket. A küldött illetve fogadott üzenetek külön szekcióban jelennek meg. Minden üzenet egy kártyán lesz látható, amit a Bootstrap implementálásával lehet használni. A kártyán szerepel a küldő, a címzett és az üzenet. Ha olyan felhasználónak küldtünk valaha üzenetet, aki ki lett törölve a rendszerből, akkor a fogadó helyén `User was deleted` szöveg jelenik meg. Minden kapott üzenetre válaszolhat a felhasználó. Ezt egy gomb segítségével teheti meg, ami a `message-sender-form.component.html` oldalt hozza elő. Ez egy nagyon egyszerű oldal, egy beviteli mezőt tartalmaz, ahova be kell írni az üzenet szövegét. Az üzenet küldése is ugyanezen az oldalon történik. A harmadik oldal, ami megjelenik az a `profile-favourites.component.html`. Itt a járművek két csoportba vannak osztva, autók és motorok. Ez az oldal egy listát jelenít meg. A lista elemei tartalmazzák a járműveket. Ezen belül minden jármű esetén egy táblázatban jeleníti meg annak adatait. A táblázat két oszlopos, de a második oszlop sorai össze vannak vonva egyetlen sorrá, ebben helyezkedik el a jármű képe. Az első oszlop sorai

tartalmazzák az adatokat, de nem az összes adatot, tartalmazza a gyártót, modellt, hengerűrtartalmat, autók esetén még azt is, hogy rendelkezik-e klímával, valamint a legvégén a céget, akihez az adott jármű tartozik.

3.7.4. Kölcsönzések, felhasználók, nyugták

A Guest jogosultságú felhasználó megtekintheti a bérléseit a `user-rents.component.html` oldalon. Ez az oldal különválasztja az autókat és a motorokat. Kártyák segítségével jeleníti meg a bérelt járműveket és azok adatait. Megjelenítésre kerül a gyártó, modell, a felhasználó, aki kölcsönözte, a cég, akitől kölcsönözte, a bérlés kezdete valamint a bérlés vége. Ha a bérlés még tart, akkor egy gombbal véget lehet annak vetni. Ha a bérlés véget ért, akkor megjelenik egy gomb, amivel ki lehet fizetni az adott járművet valamint ekkor van lehetősége értékelni a kibérelt eszközt a felhasználónak. Cég esetén a bejelentkezett cég kilistázhatja az összes kölcsönzést, amit tőle igényeltek. Ezt a `rental.component.html` oldal jeleníti meg. Admin jogosultsággal rendelkező egyének is ezen az oldalon listázzák ki a bérléseket, csak ők az összes bérlést ki tudják listázni. Az oldal felépítése megegyezik a Guest jogosultságú felhasználóknál látottéval. Az admin jogosultságú felhasználóknak van lehetőségük a felhasználók kilistázására is. Ez a `users.component.html` oldalon jelenik meg. A felhasználók adatai egy kártyán jelennek meg, mely tartalmazza az azonosítójukat, nevüket, e-mail címüket valamint jogosultságukat. Ez alatt szerepel a törlésre használatos gomb. A nyugtákat cégek és adminok is kilistázhatják. Itt lesznek láthatóak a már kész nyugták, amelyek rendelkeznek minden adattal. Ezek alatt van lehetőség a számlák újranyomtatására.

3.8. Tesztelés

Az alkalmazáson több tesztet is végrehajtottam. Minden vezérlőosztályon elvégeztem egy bizonyos számú tesztet. A tesztelés elején minden osztálynál előbb egy `@Before` művelet hajtódik végre, melyek előkészítik az osztályt a tesztelésre. Ezek nevét, feladatát illetve eredményét táblázatban mutatom be.

3.8.1. UserController tesztek

Teszteset neve	Mit tesztel	Eredmény
getUsersTestmultipleUsers	több felhasználó esetén vissza adja-e az összes egyedet	sikeres
getUsersTestemptyUsers	ha nincsenek felhasználók, üres tömbbel tér vissza	sikeres
getUsersTestsingleUser	egyetlen felhasználó esetén vissza adja-e azt	sikeres
getUserTestpresentUser	létező felhasználó megtalálása ID alapján	sikeres
getUserTestnotPresentUser	nem létező felhasználó megtalálása ID alapján -> hiba	sikeres
putUserTestExistingUser	egy létező felhasználó módosítása	sikeres
putUserTestNotExistingUser	egy nem létező felhasználó módosítása -> hiba	sikeres
deleteUserTestPresentUser	egy létező felhasználó törlése	sikeres
deleteUserTestNotPresentUser	egy nem létező felhasználó törlése -> hiba	sikeres

3.8. táblázat. UserControllerTest

3.8.2. CarsController tesztek

Teszteset neve	Mit tesztel	Eredmény
getCarsTestmultipleCars	több autó esetén vissza adja-e az összes egyedet	sikeres
getCarsTestemptyUsers	ha nincsenek autók, üres tömbbel tér vissza	sikeres
getCarsTestsingleCar	egyetlen autó esetén vissza adja-e azt	sikeres
getCartTestpresentCar	létező autó megtalálása ID alapján	sikeres
getCarTestnotPresentCar	nem létező autó megtalálása ID alapján -> hiba	sikeres
putCarTestExistingCar	egy létező autó módosítása	sikeres
putCarTestNotExistingCar	egy nem létező autó módosítása -> hiba	sikeres
deleteCarTestPresentCar	egy létező autó törlése	sikeres
deleteCarTestNotPresentCar	egy nem létező autó törlése -> hiba	sikeres

3.9. táblázat. CarsControllerTest

3.8.3. BikesController tesztek

Teszteset neve	Mit tesztel	Eredmény
getMotobikesTestmultipleMotobikes	több motor esetén vissza adja-e az összeset	sikeres
getMotobikesTestemptyMotobikes	ha nincsenek motorok, üres tömbbel tér vissza	sikeres
getMotobikesTestsingleMotobike	egyetlen motor esetén vissza adja-e azt	sikeres
getMotobiketTestpresentMotobike	létező motor megtalálása ID alapján	sikeres
getMotobikeTestnotPresentMotobike	nem létező motor megtalálása ID alapján -> hiba	sikeres
putMotobikeTestExistingMotobike	egy létező motor módosítása	sikeres
putMotobikeTestNotExistingMotobike	egy nem létező motor módosítása -> hiba	sikeres
deleteMotobikeTestPresentMotobike	egy létező motor törlése	sikeres
deleteMotobikeTestNotPresentMotobike	egy nem létező motor törlése -> hiba	sikeres

3.10. táblázat. BikesControllerTest

3.8.4. RentController tesztek

Teszteset neve	Mit tesztel	Eredmény
getRentsTestmultipleRents	több kölcsönzés esetén vissza adja-e az összes egyedet	sikeres
getRentsTestemptyRents	ha nincsenek kölcsönzés, üres tömbbel tér vissza	sikeres
getRentsTestsingleRent	egyetlen kölcsönzés esetén vissza adja-e azt	sikeres
getRentTestpresentRent	létező kölcsönzés megtalálása ID alapján	sikeres
getRentTestnotPresentRent	nem létező kölcsönzés megtalálása ID alapján -> hiba	sikeres
postRentTestSimpleBikeRent	egy kölcsönzés létrehozása motorral	sikeres
postRentTestSimpleCarRent	egy kölcsönzés létrehozása autóval	sikeres
putRentTestNotExistingRent	egy nem létező kölcsönzés módosítása -> hiba	sikeres
putRentTestExistingRent	egy létező kölcsönzés módosítása	sikeres
deleteRentTestPresentRent	egy létező kölcsönzés törlése	sikeres
deleteRentTestNotPresentRent	egy nem létező kölcsönzés törlése -> hiba	sikeres

3.11. táblázat. RentControllerTest

3.8.5. ReceiptController tesztek

Teszteset neve	Mit tesztel	Eredmény
getReceiptTestmultipleReceipt	több nyugta esetén vissza adja-e az összes egyedet	sikeres
getReceiptTestemptyReceipts	ha nincsenek nyugták, üres tömbbel tér vissza	sikeres
getReceiptTestsingleReceipt	egyetlen nyugta esetén vissza adja-e azt	sikeres
getReceiptTestpresentReceipt	létező nyugta megtalálása ID alapján	sikeres
getReceiptTestnotPresentReceipt	nem létező nyugta megtalálása ID alapján -> hiba	sikeres
postReceiptTestSimpleBikeReceipt	egy nyugta létrehozása motorral	sikeres
postReceiptTestSimpleCarReceipt	egy nyugta létrehozása autóval	sikeres
deleteReceiptsTestPresentReceipts	egy létező nyugta törlése	sikeres
deleteReceiptsTestNotPresentReceipts	egy nem létező nyugta törlése -> hiba	sikeres

3.12. táblázat. ReceiptControllerTest

3.8.6. FavouritesController tesztek

Teszteset neve	Mit tesztel	Eredmény
getFavouritesTestmultipleFavourites	több kedvenc esetén vissza adja-e az összes egyedet	sikeres
getFavouritesTestemptyFavourites	ha nincsenek kedvencek, üres tömbbel tér vissza	sikeres
getFavouritesTestsingleFavourites	egyetlen kedvenc esetén vissza adja-e azt	sikeres
getFavouritesTestpresentFavourites	létező kedvenc megtalálása ID alapján	sikeres
getFavouritesTestnotPresentFavourites	nem létező kedvenc megtalálása ID alapján -> hiba	sikeres
postFavouritesTestSimpleBikeFavourites	egy kedvenc létrehozása motorral	sikeres
postFavouritesTestSimpleCarFavourites	egy kedvenc létrehozása autóval	sikeres

3.13. táblázat. FavouritesControllerTest

3.8.7. MessagesController tesztek

Teszteset neve	Mit tesztel	Eredmény
getMessagesTestmultipleMessages	több üzenet esetén vissza adja-e az összes egyedet	sikeres
getMessagesTestemptyMessages	ha nincsenek üzenetek, üres tömbbel tér vissza	sikeres
getMessagesTestsingleMessages	egyetlen üzenet esetén vissza adja-e azt	sikeres
getMessagesTestpresentMessages	létező üzenet megtalálása ID alapján	sikeres
getMessagesTestnotPresentMessages	nem létező üzenet megtalálása ID alapján -> hiba	sikeres

3.14. táblázat. MessagesControllerTest

4. fejezet

Összegzés

Szakdolgozatomat sikeresen el tudtam készíteni, annak minden előre tervezett funkciójával együtt. A megvalósítás nem ment zökkenőmentesen. A tervezési fázisban elképzeléseimet a megvalósítás és fejlesztés során sokszor megváltoztattam illetve kibővítettem, hogy minél egyszerűbb és érthetőbb legyen a program. Az adatbázis is több táblából áll, mint ahogy azt terveztem a szakdolgozatom elején. A végleges felhasználói felület is más, mint az eredeti elképzelés volt. A szakdolgozat elkészítése során rengeteg időt töltöttem azzal, hogy dokumentációkat olvastam arról a nyelvről, amit éppen alkalmaztam, ez által jobban megismertem az adott nyelveket és kibővítettem tudásomat. Az Angular és Spring keretrendszerekbe is nagyobb betekintést nyertem ez által és közelebbről megismerkedtem velük. Nagy kedvencem a Bootstrap nyújtotta felhasználói élmények voltak. Nagyon megszerettem ezeket, mert sokkal érdekesebbé tették az alkalmazás felhasználását és kinézetét. A forráskód tárolásához és nyomon követéséhez a Git verziókezelő-rendszert használtam, és volt alkalmam megtapasztalni, milyen fontos a gyakori commit-olás és a kifejtett, jól érthető commit message-ek használata annak érdekében, hogy a munka menete később is jól áttekinthető maradjon. Ez számomra nagy újdonságot jelentett, habár már használtam verziókezelőt, de csak szerkesztőként voltam jelent a projectben, és nem én tartottam karban a projectet. Most viszont ez a feladat is rám hárult, aminek eredménye az lett, hogy sokat olvastam a Git adta lehetőségekről is. A dolgozat szövegét és a táblázatokat LaTeX segítségével készítettem el, melynek alapjait szintén időközben tudtam elsajátítani. A szakdolgozat elkészítése folyamán igyekeztem arra törekedni, hogy minél egyszerűbb felhasználói felületet biztosítsak a felhasználók

számára, mivel az alkalmazást úgy készítettem el, hogy nap mint nap használhassák, és mivel elsősorban turisták számára készült, ezért nem lenne előnyös, ha túl lenne bonyolítva, és a felhasználónak segítséget kellene kérnie, ami akár nehézség is lehet egy idegen nyelven, mint a felhasználónak, mint a rendelkezésre álló személynek. Egyszerű és könnyen kibővíthető kódot írtam szerintem, amit könnyen lehet értelmezni. Igyekeztem minél több dokumentálást belevinni a kódolásba, hogy növeljem annak értelmezhetőségét. Úgy vélem, hogy a munkám a felhasználók számára hasznosnak fog bizonyulni, az alkalmazást tovább fejlesztőknek pedig nem okoz majd gondot az általam elkészített kód továbbfejlesztése.

4.1. Fejlesztési lehetőségek

Google és Facebook használata bejelentkezéshez

Egy hasznos fejlesztés lenne, ha a felhasználóknak lehetőségük lenne bejelentkezni Facebook vagy Google fiókkal.

A járművek rendelkezzenek képpel

Sokat segíthet a felhasználóknak a jármű kiválasztása során, ha láthatják, hogy hogyan is néz ki a jármű.

Nyugta kinézetének módosítása

Attól függően, hogy melyik cég fogja használni az elkészült alkalmazást, a nyugtába fel kellene tüntetni a cég adatait, esetleg logóját, és minden hivatalos adatot, ami szükséges egy igazi, valid számla előállításához.

A cégek profiljának megtekintése

Lehetne egy olyan továbbfejlesztés, ahol a felhasználók megtekinthetnék a cégek profiljait.

Irodalomjegyzék

[1] Java dokimentáció SE 8. *Java dokumentáció*. https://docs.oracle.com/javase/8/docs/api/. Elérés dátuma: 2019-12-08.

[2] Spring Boot. *Spring boot*. https://spring.io/projects/spring-boot. Elérés dátuma: 2019-12-08.

[3] Angular dokumentáció. *Angular dokumentáció*. https://angular.io/docs. Elérés dátuma: 2019-12-08.

[4] *Bootstrap dokumentáció*. https://getbootstrap.com/docs/4.3/about/overview/. Elérés dátuma: 2019-12-08.

[5] Horváth Győző és Tarcsi Ádám. *Webadatbázis-programozás*. http://ade.web.elte.hu/wabp/lecke7_lap1.html. Elérés dátuma: 2019-05-01.

[6] *ng-starrating dokumentáció*. https://www.npmjs.com/package/ng-starrating. Elérés dátuma: 2019-12-08.

[7] *ngx-toastr dokumentáció*. https://www.npmjs.com/package/ngx-toastr. Elérés dátuma: 2019-12-08.

Ábrák jegyzéke

Táblázatok jegyzéke